APPLICATION OF VIRTUAL PRIVATE NETWORKING TECHNOLOGY TO STANDARDS-BASED MANAGEMENT PROTOCOLS ACROSS HETEROGENEOUS FIREWALL-PROTECTED NETWORKS

Steven O'Guin, Chris K. Williams, Nikolaos Selimis
Defense Information Systems Agency Joint Interoperability and Engineering Office
Booz-Allen & Hamilton McLean, Virginia

ABSTRACT

The past two decades have seen tremendous growth within DoD of enterprise-wide COTS-based messaging and communications systems, including the Defense Message System, the Global Command and Control System, and the Global Combat Support System. To economize on development costs, standards-based protocols – including SMTP, SNMP, FTP, Telnet, and HTTP – are used to implement the underlying functionality of these systems, including messaging and service management.

During the past five years, vulnerabilities in such standards-based protocols have been identified, and security over the Internet and its connected systems has become an ever-increasing concern. Network security policies have been created to address the dilemma of protecting local systems from external attack while permitting easy communications between authorized parties. A burgeoning industry of firewall manufacturers has arisen to meet the challenge of implementing these policies effectively, safely, and reliably. Virtual Private Networking (VPN) technology was developed to enable separate firewall-protected enclaves to safely exchange data over unsecured networks. This technology is still maturing and standardized – using IPSec, ISAKMP, and DES encryption – to enable separate VPN implementations to interoperate over shared networks.

This paper studies how Virtual Private Networking technology can be employed to protect the use of standards-based service management protocols – including FTP, Telnet, SNMP, and NTP – across heterogeneous firewall-protected networks, balancing the requirements of enterprise service management with the need for local-level network security.

INTRODUCTION

The capability underlying all VPN implementations is a software concept known as protocol tunneling. Tunneling is a mechanism by which an application's site-to-site communications are protected from unauthorized access by encasing them within the transmission protocols of a completely different application. Using this technique, the original data stream can also be encrypted and authenticated to protect it against unauthorized viewing or modification.

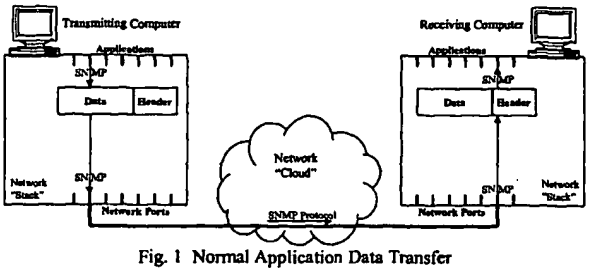


Fig. 1 Normal Application Data Transfer

In normal network application data transfer, an application on the transmitting computer opens a "port" to the receiving computer using either the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP). Next, the computer sends its data across the network to the receiving computer using the data format of the application protocol. In the example shown in Fig. 1, the following process is observed: (1) Simple Network Transfer Protocol (SNMP) data is sent from the originating application through the transmitting computer's network stack to a randomly selected UDP output socket. (2) From the transmitting computer's output socket, the data is then sent "in the clear" over the Wide-Area Network (WAN) to the receiving computer's well-known SNMP UDP input socket. (3) The receiving computer then passes the SNMP data back up through the network stack to its own SNMP application. Because this data is formatted using a well-known standards-based protocol, and then sent over the WAN "in the clear", a hacker monitoring the network can easily see the data being transmitted, understand its meaning, and then manipulate it to attack either the sender's or the recipient's computers. Standards-based management protocols – including FTP, Telnet, and SNMP – are extremely vulnerable to such attack when they are transmitted over the Internet, since their formats are well-known, and they provide only minimal protection to their communications.

When tunneling is employed to protect the underlying protocol, an additional step is introduced to the network communications process, as shown in Fig. 2: (1) The original data packets from the application on the transmitting computer are intercepted in the network stack by protocol tunneling software. (2) The tunneling software then encodes the data and header information of the original packets and places the encoded packets into the data portion of new packets formatted using a different, and possibly customized, protocol. To provide additional protection, the original

0-7803-5538-5/99/$10.00 © 1999 IEEE

protocol's data packets may also be digitally signed and/or encrypted before being placed within the tunnel protocol. (3) The tunnel protocol's packets have new headers that route them over the WAN to the tunnel protocol's network socket on the receiving computer, instead of to the original protocol's network socket. (4) At the receiving computer, the tunnel packets are received at the tunnel's network socket, and the protocol tunneling software on the receiving computer intercepts them and decodes the original encapsulated application packets. (5) These decoded packets are then passed up the network stack to the receiving application, which receives the data normally; completely unaware of the tunneling process that has just taken place.

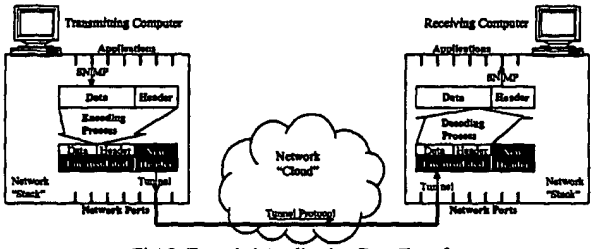


Fig. 2 Tunneled Application Data Transfer

When protocol tunneling is used, the only packets that travel over the WAN are tunneled packets that use an uncommon, and possibly secured protocol. This makes it difficult for hackers to identify, decode, and modify the encapsulated application protocol packets, especially if the tunnel software signs or encrypts them before transmission. Since the encoding and decoding processes are performed within the network stacks of the sending and receiving machines invisibly to the application software, standards-based applications can be used securely without requiring costly application programming changes. This "plug-and-play" approach also gives integrators a great deal of flexibility to choose security systems separately from applications, and to easily integrate new applications to take advantage of existing protocol-tunneling based security systems.

This versatility comes at a price, however. Because protocol-tunneling components – either software, hardware, or otherwise – would need to be installed throughout the enterprise to protect the service management system, they must be fully compatible with all of the system's components and intercommunications protocols. Furthermore, because protocol tunneling can enclose multiple application protocols within a single tunnel protocol, local site network security policies that are enforced by their firewalls must be carefully crafted to include policies for handling tunneled application protocols differently from their "in the clear" versions. The addition of protocol tunneling makes it possible to securely use management protocols that are not secure when used in the clear, and network security policies must be adjusted to allow for this fact. This management of the protocol tunnel's security policy may require cooperation from both ends of the protocol tunnel to ensure that security leaks cannot occur.

VPN ARCHITECTURE

Virtual Private Networking involves applying protocol tunneling techniques to protect the WAN communications between multiple firewall-protected enclaves, creating a protected "virtual" network that makes use of the underlying networking hardware and software for connectivity. The only network traffic that is visible on the WAN is the encrypted and/or authenticated VPN traffic, making it unreadable and unmodifiable by unauthorized users. This protection can be applied to all commonly-used Internet protocols, including Simple Mail Transfer Protocol (SMTP), Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Network Time Protocol (NTP), Simple Network Management Protocol (SNMP), and Telnet. The architecture of a typical VPN is shown in Fig. 3.

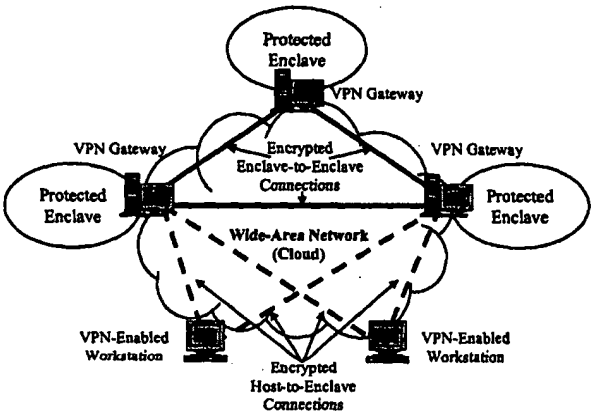


Fig. 3 VPN System Architecture

VPN implementations generally consist of two types of components: a VPN gateway that is installed to serve all of the computers at a single site, and a VPN client that enables a single host workstation to connect to the VPN. The VPN gateway is typically a stand-alone system that enables both enclave-to-enclave and enclave-to-host VPN-based communications outside the enclave. This functionality may be combined with the enclave's firewall, enabling one network-hardened component to serve both functions. The VPN client is a separate software package that is installed on conventional workstations to enable them to establish secure host-to-enclave communications links with the VPN gateway components via the WAN.

USING VPN PROTECTION

This VPN technology can be "bolted on" to an existing enterprise system to provide additional protection to its standards-based management protocols. Large-scale enterprise information systems typically consist of two sections.

First, there is a *backbone* consisting of *Control Centers* and *Data Centers* that are operated by the system's proponent, typically a corporation or government agency. Second, there are a large number of *Local Sites* that have clients that interact with the backbone for data access and remote management. Fig. 4 illustrates such an architecture, and shows how VPN technology could be used to protect the service management communications between the Control Center, the Data Centers, and the Local Sites.

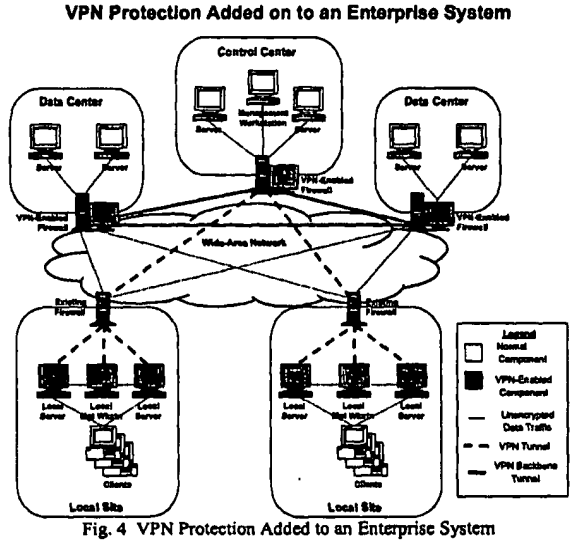


Fig. 4 VPN Protection Added to an Enterprise System

Typically, it is practical to add custom components to the backbone sites, but not economically or politically feasible to drastically re-configure the local sites in such an architecture. For this reason, the architecture shown in the figure uses firewalls with integrated VPN gateways to create a VPN between the backbone sites that protects their service management communications as they travel over the WAN, while leaving the existing local site firewalls alone.

Once the backbone's communications are protected, the next step is to provide VPN-based protocol protection from the backbone's VPN to the components at the local sites that with which the backbone interacts. Since the VPN provides for the authentication and transport security that the original protocols lacked, it may be acceptable to local site security authorities to permit service management protocols to enter the local sites if they are protected by the VPN. There are three techniques that can be used to achieve VPN connectivity from the local sites to the backbone.

1. When the local sites' existing firewalls have built-in VPN capabilities that are compatible with the backbone's VPN gateways, enclave-to-enclave VPN connections can be created between the local sites and the backbone.
2. When the local sites' existing firewalls are not capable of interoperating with the backbone VPN, standalone VPN gateways can be installed behind local site firewall to provide VPN connectivity for local site components.
3. Alternatively, when the local sites' existing firewalls are not capable of interoperating with the backbone VPN, VPN client software can be installed on local site computers that have to interact with the backbone, and they create VPN tunnels through the local firewalls to the backbone.

The remainder of this paper considers the third approach, in which VPN client software is installed directly onto local components so that they can communicate through the local firewall with the backbone. Because most VPN client software is designed to be "plug and play", it should be possible to install it onto existing platforms without disrupting the operation of the existing application software. To fully analyze the operation of this architecture, the Local Area Networks (LANs) at the backbone sites and the local sites require closer inspection.

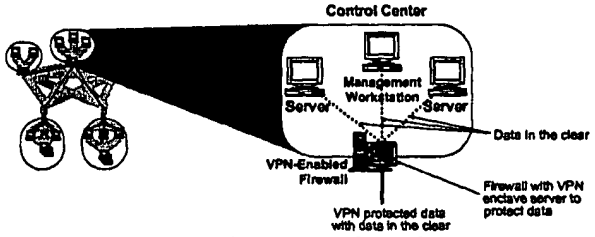


Fig. 5 VPN Connection at a Control Center

At a Control Center, VPN connectivity is provided through a single VPN gateway, which may be integrated with the site firewall, as illustrated in Fig. 5. The system components connected to the site's LAN send and receive all of their network traffic in the clear, with no additional protection. However, when that network traffic must leave the LAN and traverse the WAN, it passes through the site's VPN gateway, which examines the traffic. If it is destined for another backbone site, or if it is destined for a VPN-enabled local site or local site component, it is encrypted by the gateway and then sent out via a VPN tunnel, protecting it from attack.

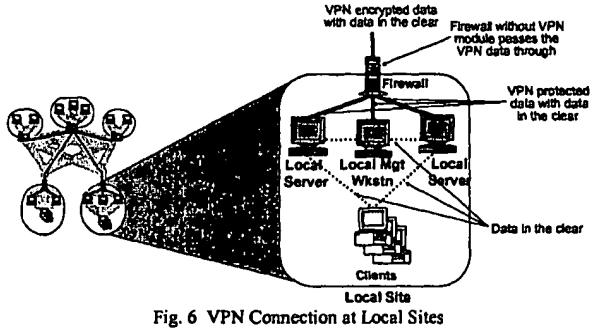


Fig. 6 VPN Connection at Local Sites

At local sites that do not have a VPN-enabled firewall or standalone VPN gateway, the tunneled traffic passes through the local firewall to the individual VPN-enabled components

located at the site. The local firewall must be configured to permit the VPN protocols to pass through, which raises network security policy issues that will be considered later in this paper. At the VPN-enabled components, the VPN client software intercepts and decrypts the VPN traffic at the network layer, passing the unencrypted data up the network stack to the system applications, which can operate unaware that VPN protection took place at all. It is important to note that this architecture does not require that all local systems have the VPN client software installed – only those components that must communicate directly with the enterprise system's backbone. Other communications within the local sites, such as between system servers and end-user clients, can be conducted "in the clear."

IMPACT ON SYSTEM PROTOCOLS

Examining the protocols used by enterprise systems in more detail, one can see how they are affected by the introduction of VPN technology. Typically, service management protocols are the first candidates for VPN protection, because they are the most vulnerable to surreptitious monitoring or attack, as shown in Fig. 7. Most VPN software packages can be configured to support this need by protecting application data on a per-socket basis, permitting specific applications to be protected while others operate in the clear.

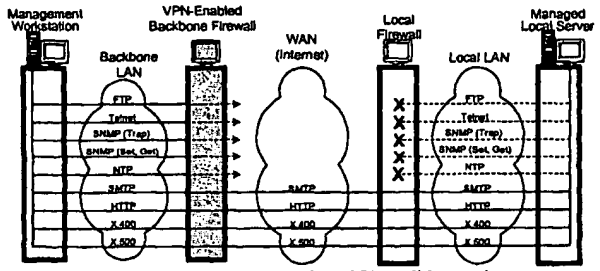


Fig. 7 Management Protocols and Firewall Interaction

The protocols of most concern are those typically used for system management: FTP, Telnet, SNMP Traps, SNMP Sets and Gets, and NTP. They are generally blocked by local site firewalls to protect against the threat that they pose to local systems and the data stored on them. A VPN is ideal for protecting these protocols from tampering as they travel over the WAN, and would have the added benefit of permitting the backbone site firewalls to be configured to block these vulnerable protocols as well. Other protocols typically used by enterprise systems – SMTP, HTTP, X.400, and X.500 – are generally permitted by local network security policies to transit the WAN and local firewalls in the clear, and do not need to be protected.

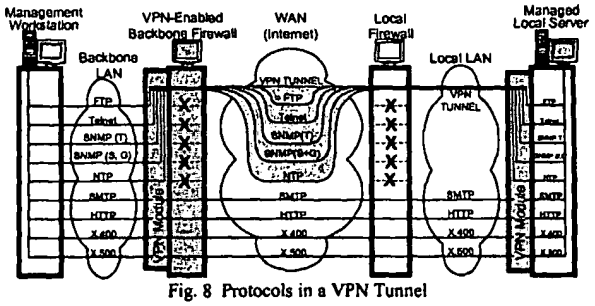


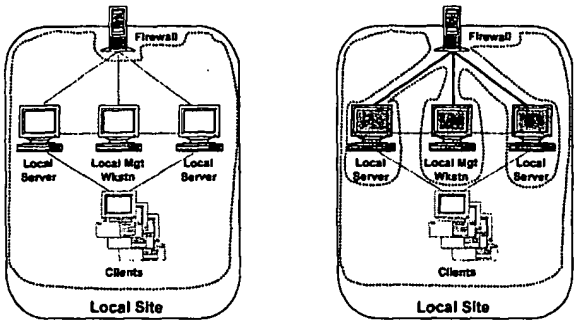
Fig. 8 Protocols in a VPN Tunnel

By employing VPN technology, all of the system management protocols can be encapsulated within a VPN tunnel between the backbone and the local sites. This tunnel may be configured differently at each end, depending on whether the site firewall, a separate VPN gateway, or a VPN-enabled component terminates the link, as shown in Fig. 8. At backbone sites, it makes logical sense for VPN capabilities to be provided by a central VPN gateway or VPN-enabled firewall. When service management protocols must pass outside the backbone site's local network, they are tunneled for transit over the WAN. Furthermore, the backbone site firewalls would be configured to prohibit the passage of those protocols outside of the VPN.

Once outside the backbone enclave, the only communications that are visible on the WAN are the VPN protocol and the "safe" communications protocols. All of the vulnerable service management protocols are encapsulated within the VPN tunnel, which may also provide additional authentication and encryption to protect them from monitoring and attack. At local site firewalls, the VPN protocol would pass through while the unprotected service management protocols are prohibited. At the VPN-enabled local components, the VPN protocol would be intercepted within the network stack, decrypted, decoded, and finally passed up the stack to the appropriate applications. The applications are ignorant of all of this processing, and operate normally, ignorant of the VPN and protocol tunneling that is protecting their communications.

POLICY ISSUES

This approach creates an interesting policy dilemma for system accreditors because the service management protocols encapsulated by the VPN tunnel can effectively bypass local network security policies as they are enforced at the firewall. Consequently, local network security policies must be adjusted to include the configuration of the VPN and VPN-enabled local components to mitigate the vulnerability of the site to outside attack via either the VPN components themselves or the system management protocols encapsulated within the VPN.



Standard Network Security Policy Implementation using Firewall

Modified Network Security Policy Accommodating VPN Tunnel

Fig. 9 VPN Tunnel Impact on Security Policy Perimeter

Fig. 9 shows the effect that VPN-enabled local components can have on a local site's network security policy domain. When local site components are VPN-enabled, they can use the VPN tunnels to communicate with the WAN using the sensitive system management protocols. Consequently, both the VPN software and the VPN-enabled component must be carefully configured to avoid creating "back doors" into the local site network that could be exploited by a VPN-enabled attacker. Through the VPN, the VPN-enabled components become part of the site's network boundary, and must be configured to ensure that the boundary's security is maintained. The local network security policy, which originally governed the configuration of only one system – the site firewall – now must be expanded to include the configuration of the VPN-enabled components as well.

For example, if the original local network security policy did not allow incoming SNMP to go through the firewall, but SNMP is part of the enterprise system's management architecture, then the VPN would have to be implemented such that the enterprise system's SNMP is permitted to reach the VPN-enabled local components, while at the same time prevented from getting past them to the rest of the local site. Only in this way can the VPN be implemented while preserving the original network security policy. Similarly, if the local-level network security policy did not permit outbound FTP, then the local VPN-enabled components would have to be configured to not propagate outbound FTP via the VPN tunnels. Only in this way can VPN tunnels be implemented without compromising the integrity of local-level network security policies.

These challenges and other considerations are encapsulated within the following questions, which must be considered by network security policy personnel prior to implementing a VPN tunneling solution:

1. Is it acceptable for components behind the firewall to have FTP, Telnet, SNMP, or NTP connections with the enterprise system's backbone sites if adequate security for those protocols can be ensured?
2. Do VPNs with PKI-based strong authentication provide adequate protection to permit the transit of FTP, Telnet, SNMP, and NTP protocols over WANs such as the Internet?
3. Is it acceptable for local-site firewalls to be configured to permit VPN protocols to pass through to VPN-enabled enterprise information system components within the local site?
4. What information is needed to determine whether VPN client software installed on local components will provide adequate security to protect the remainder of the Local Site's components from attack via the VPN-tunneled protocols?
5. What additional precautions are needed to protect the rest of the Local Site from attack via FTP, Telnet, or SNMP transmitted through the firewall to VPN-enabled local components?
6. Are these precautions adequate for local site accreditation?
7. What additional precautions need to be taken to "compartment-alize" the enterprise system's security to prevent a security breach at one site from being propagated through the VPN to other sites?

CONCLUSION

This approach provides a viable alternative to re-engineering standards-based enterprise information systems in order to increase the security of their service management capabilities. Before it can be implemented on an enterprise information system, however, significant policy agreements must be made between the network security organizations of the system proponent and the local sites that would implement it. Furthermore, the technical capabilities of actual VPN products would have to be validated in a laboratory environment to confirm their suitability for implementation at an enterprise scale. Accounting for all of these issues, it is possible for robust and flexible enterprise-wide service management to be successfully balanced with the equally pressing need for strong, compartmented, enterprise-wide network security.

REFERENCES

[1] DoD Office of the Secretary of Defense, "Security and technical requirements for firewalls," June, 1998.
[2] U.S. Army Director of Information Systems for Command, Control, Communications and Computers, "Army firewall and high assurance guard implementation guidance," December, 1998.
[3] Lockheed-Martin Federal Systems, "Service management strong authentication," November, 1997.
[4] Information Security Magazine, "IP: the next generation," February, 1999.
[5] Raptor Systems, "Eagle 5.0 firewall," 1997.
[6] Secure Computing Corporation, "SecureZone technical brief," July, 1998.
[7] Cisco Systems, "IPSec white paper," September, 1998.
[8] Ken Masica, (Sun Microsystems) "SKIP your way to security," June 1997.
[9] Jonathan Chinitz and Steve Sonnenberg, (Intellisoft) "A transparent security framework for TCP/IP and legacy applications," August, 1996.
[10] CheckPoint Technologies, "Redefining the virtual private network," 1998.